REMARKS

Initially, in the Office Action dated October 27, 2003, the Examiner objects to the specification because of informalities. The title has been objected to as lacking descriptiveness. Figs. 1-4 have been objected to. The Examiner has required more IDS information under 37 C.F.R. §1.105 and asserts that the Office Action had an attached requirement for information under 37 C.F.R. §1.105, however, the Office Action contained no such attachment.

Claims 1-5 have been objected to because of informalities. Claims 1, 4 and 5 have been rejected under 35 U.S.C. §112, first paragraph. Claims 1-5 have been rejected under 35 U.S.C. §112, second paragraph. Claims 1, 2, 4 and 5 have been rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,453,468 (D'Souza). Claim 3 is rejected under 35 U.S.C. §103(a) as being unpatentable over D'Souza in view of U.S. Patent No. 5,897,658 (Eskesen).

By the present response, Applicant has submitted a substitute specification and has amended Figs. 1-4 and claims 1-5 to further clarify the invention. Claims 1-5 remain pending in the present application.

Specification Objections

The disclosure has been objected to because of informalities. Applicant has amended the disclosure and is submitting a substitute specification as required by the Examiner. Accordingly, Applicant respectfully requests that these objections be withdrawn.

The Examiner has required a new title of the invention asserting that the current title is not descriptive. Applicant has submitted a new title of the invention in accordance with the Examiner's suggestion.

Drawings Objections

Figures 1-4 have been objected to. Applicant has amended the Figures to further clarify the invention and respectfully request that these objections be withdrawn.

The Examiner has required new corrected drawings asserting that Fig. 1 does not convey information to enable any person skilled in the art and cites terms "start", "stop", "change information" and "management object". Applicant is confused as to the Examiner's requirement. Applicant submits that Fig. 1 fully enables a person of ordinary skill in the art at the time of the invention to practice the claimed invention. The "start" and "stop" signals travel between the factory object 20 and the existing server objects and new server objects. The "change information" is information sent between server objects and factory object 20 and management object 10. The functionality of Fig. 1 is fully disclosed in Applicant's specification on pages 6-11. Applicant has made a minor amendment to Fig. 1 to further clarify the invention. Applicant can supply the Examiner with more information to aid the Examiner's understanding if necessary. Accordingly, Applicant respectfully requests that these objections be withdrawn.

IDS Requirements

The Examiner has required Applicant to provide information required under 37 C.F.R. §1.105. The Examiner states that a requirement for information under 37 C.F.R. §1.105 was attached to the Office Action, however, Applicant found no such attachment with the current Office Action. In the Office Action, the Examiner states that in response to this requirement that Applicant is to provide the title, citation and copy of each publication that is a source used for the description of the prior art in the disclosure and a concise explanation of that publication's contribution to the description of the prior art. Applicant submits that information such as this has already been supplied in Applicant's disclosure under Background of the Invention where prior art reference JPA 11-154138 is fully described along with the problems associated with the method described in this reference. In addition, Applicant is filing concurrently with this response an IDS and 1449 with a reference in response to this requirement to further aid the Examiner. Any other information under this requirement is unknown or is not readily available to the Applicant.

Claim Objections

Claims 1-5 have been objected to because of informalities. The Examiner asserts that claim 1, 4 and 5 contain the term "server object information acquisition unit" and that this term is confusing. Applicant submits that this term as recited in the claims of the present application is not confusing and clearly described in Applicant's specification and drawings. For example, in the embodiments of the present invention shown in Figs. 4 and 5 it is shown that a “server object information acquisition unit" may, depending on the embodiment, may reside at a client or a server. Accordingly, Applicant respectfully requests that these objections be withdrawn.

35 U.S.C. §112 Rejections

Claims 1, 4 and 5 have been rejected under 35 U.S.C. §112, first paragraph. The Examiner asserts that the term "said access" in claims 1, 4 and 5 lacks sufficient antecedent basis. Applicant respectfully disagrees and asserts that this term has antecedent basis in each claim in the line preceding where it is stated "accessing a server object . . . ." Accordingly, Applicant respectfully requests that these objections be withdrawn.

Claims 1-5 have been rejected 35 U.S.C. §112, second paragraph. Applicant has amended these claims to further clarify the invention. Applicant submits that these terms are nouns, the details of which are fully disclosed in the specification and claims of the present application. In this regard, Applicant asserts that these claims are not indefinite. Accordingly, Applicant respectfully requests that these rejections be withdrawn.

35 U.S.C. §102 Rejections.

Claims 1, 2, 4 and 5 have been rejected under 35 U.S.C. §102(e) as being anticipated by D'Souza. Applicant respectfully traverses these rejections.

D'Souza discloses enhancing reliability while upgrading a software program implemented in a clustered computer system from a first version to a second version. A software program is implemented as software modules running on a plurality of computers coupled in a cluster configuration in a clustered computer system.. A certification level associated with each of the software modules is checked and if the software module has a first certification level, a load level on the given software module is limited to a first load level, and if the certification level on a given software module has a second certification level, then the load level on the second routing transaction request is allowed to reach a second load level higher than the first load level.

Regarding claims 1, 4 and 5, Applicant submits that D'Souza does not disclose or suggest the limitations in the combination of each of these claims of, inter alia, requesting server object information of server objects to be accessed, selecting and supplying a requestor with server object information of a newest server object of requested server objects based on a change information including revision information showing a newness of each of the requested server objects where the newest server object is the most recently accessed server object of the requested sever objects, accessing a server object indicated in the server object information supplied, or performing processing requested by the access. The Examiner asserts that D'Souza discloses requesting server object information of server objects to be accessed at col. 2, lines 32-46. However, this portion of D'Souza merely discloses forwarding a request containing a URL to a webserver router, which arbitrates among other webservers to decide which of the webservers should service the user's request. This is not requesting server object information of server objects to be accessed, as recited in the claims of the present application. A webserver router arbitrating among webservers based on a request containing a URL has nothing to do with requesting server object information of server objects to be accessed. There is no request of server object information disclosed in Dsouza, just a decision on which webserver should handle a request.

The Examiner further asserts that D'Souza discloses selecting and supplying a requestor with server object information at col. 7, lines 1-20. However, this portion of D'Souza merely discloses the operation of a software module being monitored to decide whether the reliability of the software module satisfies predetermined material or not. This is not selecting and supplying a requestor with server object information of a newest server object of requested server objects based on a change information including a revision information showing a newness of each of the requested server objects where the newest server object is the most recently accessed server object of the server objects, as recited in the claims of the present application. D'Souza merely discloses upgrading a software program by first ascertaining a certification level associated with a software module and upgrading the software module according to the certification level. D'Souza does not disclose or suggest anything related to a newest server object of requested server objects, revision information showing a newness of each requested server objects, or which of server objects to be accessed is the server object most recently accessed.

The Examiner further asserts that D'Souza discloses accessing a server object indicated in the server object information supplied and performing processing requested by the access at col. 7, lines 52-56. However, this portion of D'Souza merely discloses that an IDA is included with each of the clusters that implement the webserver stage, the business logic stage, and the data repository stage. This does not disclose or suggest anything related to accessing a server object indicated in server object information supplied, where, as noted previously, the server object information supplied includes a most recently accessed server object of the requested server objects. Moreover, this portion of D'Souza does not disclose or suggest performing processing requested by the access, as recited in the claims of the present application. D'Souza discloses that IDAs implement respective stages and thus have no connection with an access request, as disclosed in Applicant's invention. Further, in D'Souza, load level is raised if the reliability of software modules becomes higher to a certain extent, however, D'Souza does not return information concerning a revision change of server objects to a requestor, as disclosed in the claims of the present application.

Regarding claim 2, Applicant submits that this claim is dependent on independent claim 1 and, therefore, is patentable at least for the same reasons noted regarding this independent claim. For example, D'Souza does not disclose or suggest selecting and supplying the requestor with the server object information of the server object to be accessed in accordance with the change information of the requestor server object.

Accordingly, Applicant submits that D'Souza does not disclose or suggest the limitations in the combination of each of claims 1, 2, 4 and 5 of the present application. Applicant respectfully requests that these rejections be withdrawn and that these claims be allowed.

35 U.S.C. §103 Rejections

Claim 3 has been rejected under 35 U.S.C. §103(a) as being unpatentable over D'Souza in view of Eskesen et al. Applicant respectfully traverses the rejection.

Eskesen et al. discloses protecting portions of memory by providing access requests to a communications area for processing by a hidden server. A memory in a computer system includes a visible portion and a hidden portion. The visible portion of the memory is addressable by a processor and the operating system operating within the computer system. Addressability by either the processor or the operating system is excluded to the hidden portion of the memory. A hidden server which addresses both the visible portion of memory and the hidden portion of memory receives requests for data access from the communications area and initiates data access from the hidden memory.

Applicant submits that claim 3 is dependent on independent claim 1 and, therefore, is patentable at least for the same reasons noted regarding this independent claim. Applicant submits that Eskesen et al. does not overcome the substantial defects noted previously regarding D'Souza. For example, Applicant submits that neither none of the cited references disclose or suggest stopping server objects having old change information including old version information in a case where there are a plurality of server objects having a same server object name or same interface identification information. Eskesen et al. merely discloses that descriptors produced by old versions of services are allowed to be understood by the new version of services. Eskesen et al. does not disclose or suggest invalidation of server objects having old versions, as recited in the claim of the present application.

Accordingly, Applicant submits that none of the cited references, taken alone or in any proper combination, disclose, suggest or render obvious the limitations in the combination of claim 3 of the present application. Applicant respectfully request that these rejections be withdrawn and that this claim be allowed.

In view of the foregoing amendments and remarks, Applicant submits that claims 1-5 are now in condition for allowance. Accordingly, early allowance of such claims is respectfully requested.

To the extent necessary, Applicants petition for an extension of time under 37 CFR 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, or credit any overpayment of fees, to the deposit account of Antonelli, Terry, Stout & Kraus, LLP, Deposit Account No. 01-2135 (referencing attorney docket no. 500.40188X00).

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

Frederick D. Bailey
Registration No. 42,282

FDB/sdb
(703) 312-6600

Attachments: Replacement Sheet
Annotated Sheet Showing Changes
Substitute Specification
Marked-up Copy of Specification

~~DISTRIBUTED OBJECT MANAGEMENT METHOD, IMPLEMENTATION SYSTEM AND RECORDING MEDIUM FOR RECORDING THE PROCESSING PROGRAM FOR THE METHOD~~ AN IMPROVED SYSTEM AND METHOD TO DYNAMICALLY SELECT AND LOCATE SERVER OBJECTS BASED ON VERSION INFORMATION OF THE SERVER OBJECTS

CROSS-REFERENCE TO RELATED APPLICATIONS

Not Applicable

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a distributed object management system for managing objects in a distributed object environment, and more specifically to dynamically selecting and locating server objects, in a distributed object environment, based on version information of the server objects.

Description of the Related Art

In a distributed object environment for executing a backbone job, demand is now high for realizing operation around the clock for 365 days a year. In such a distributed object environment operated continuously, it is indispensable to carry out the change of a server object, if required, in a dynamic way.

For the version-up of a server object, for example, JP-A-11-

154138 describes a method in which a server has a backup file of a load module so that the internal information of the load module and the internal information of the backup file of the server to be started are checked at the time of start up.

In the conventional distributed object environment described above, even in the case where a new server object is started to provide the same service as other server objects, it is difficult to carry out the change while a plurality of the server objects are in operation as long as the object name or the interface of the new server object remains unchanged in view of the fact that a plurality of server objects providing the same service are considered to be equivalent to each other.

Another problem of the method described in JP-A-11-154138 is that in the case where a version-up is required, the start-up is notified to clients as a failure, and therefore the service is interrupted. Further, in the distributed object environment, a problem is posed that version consistency fails to be considered between a plurality of server objects which may be operated in collaboration with each other.

In the case where the service of a server object is provided continuously or a plurality of server objects are operated in collaboration with each other in a distributed object environment, therefore, it is necessary that a server object can be changed while retaining consistency between the server objects in operation.

BRIEF SUMMARY OF THE INVENTION

The object of the present invention is to solve the problems described above, and to provide a technique capable of changing a server object in operation.

According to one aspect of the invention, there is provided a distributed object management system for managing objects in a distributed object environment, wherein in the case where the server object information is requested, the server object information of the newest server object based on change information is supplied to the requester to enable the newest server object to be accessed. The newest server object being the server object most recently accessed. In this way, the server object accessed is changed from an old one to the newest one.

According to another aspect of the invention, there is provided a distributed object management system, wherein the change information indicating the newness of the server object is held, and the change of a given server object is identified by determining the difference of this change information, thereby making it possible to change an object in operation.

According to still another aspect of the invention, in the case where a requester client object or a server object requests the server object information of a server object or another server object, as the case may be, to be accessed, the change information of the requested server objects are checked, so that the newest one of the server objects is selected and the server object information of the particular server object is supplied to the requester.

The requester that has received the server object information of the newest server object accesses the server object indicated in the supplied server object information and requests the execution of the processing thereof. The server object thus accessed executes the requested processing.

In the case where there exist a plurality of server objects having the same server object name or the same interface identification information, the change information of these server objects are compared with each other, and the server object having old change information is canceled.

As described above, with a distributed object management system according to this invention, in the case where the server object information is requested, the server object information of the newest server object is supplied to the requester to enable the newest server object to be accessed. In this way, the server object accessed is changed from an old one to the newest one. Therefore, a server object can be changed in operation.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing a configuration of a distributed system according to an embodiment of the invention.

Fig. 2 is a diagram showing an example of configuration of a management object 10 according to an embodiment of the invention.

Fig. 3 is a diagram showing an example of configuration of a factory object 20 according to an embodiment of the invention.

Fig. 4 is a diagram showing an example of configuration of a client object 30 according to an embodiment of the invention.

Fig. 5 is a diagram showing an example of configuration of a server object according to an embodiment of the invention.

Fig. 6 is a diagram showing an example of configuration of the server object registration information used by the management object 10 in a server object information management unit 10a and a server object information select unit 10b.

Fig. 7 is a diagram showing an example of configuration of the server object management information used in the server object life cycle management unit 20a of the factory object 20 according to an embodiment of the invention.

Fig. 8 is a flowchart showing the steps of the process in the server information management unit 10a of the management object 10 according to an embodiment of the invention.

Fig. 9 is a flowchart showing the steps of the process in the server object information select unit 10b of the management object 10 according to an embodiment of the invention.

Fig. 10 is a flowchart showing the steps of the process in the server object life cycle management unit 20a of the factory object 20 according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE ~~EMBODIMENTS~~INVENTION

An explanation will be given below of a distributed object management system according to an embodiment of the invention, in which an object is changed in operation.

Fig. 1 is a diagram showing a configuration of a distributed system according to this embodiment. As shown in Fig. 1, the distributed system according to this embodiment comprises a management object 10, a factory object 20, a client object 30, an existing server object A 40, an existing server object B 50 accessed by the existing server object A 40, a new server object A 60 started after the factory object 20 to change the existing server object A 40 and the existing server object B 50, a new server object B 70, change information 52 delivered to the management object 10 and the factory object 20 upon start (23) and registration (41) of the existing server object A 40, and change information 72 delivered to the management object 10 and the factory object 20 upon start (24) and registration (71) of the new server object B 70. The objects are executed by different information processing systems, respectively, and communicate with each other through a network to transmit/receive information associated with start/stop instructions, registration and retrieval.

In the distributed system according to this embodiment, assume that the process for the server object A and the server object B is carried out in response to the access from the client object 30. First, the factory object 20 starts (21) the management object 10, starts (22) the existing server object B 50 and starts (23) the existing server object A 40. The factory object 20 starts the existing server object B 50 and the existing server object A 40 in that order, i.e. in the predefined reverse order of access to the server objects.

The existing server object A 40 and the existing server object B 50, once started by the factory object 20, send the server object information to the management object 10. The positional information such as an IP (internet protocol) address indicating the position, the ID information including the name of the server object and indicating the interface, and the change information indicating the newness of the server objects are registered in the management object 10. The change information includes the revision information, etc. for determining the newness of a plurality of the server objects providing the same service and is arbitrary information that can be determined in this particular distributed system. In the case where a given server object has no function to transmit the change information, the management object 10 registers information such as the present time of registration of the particular server object as change information.

Now, an explanation will be given of the operation for executing the client object 30 in the distributed system of this configuration. The client object 30 requests the management object 10 to acquire the server object information of the server object A accessed by the client object 30.

The management object 10, upon receipt of the request from the client object 30 to acquire the server object information of the server object A, retrieves the server object information of the server object A, acquires the server object information of the existing server object A 40 already started as a server object A and supplies it to the requester client object 30 (31).

The client object 30, upon receipt of the server object information of the existing server object A 40 from the management object 10, accesses the existing server object A 40 using the particular server object information (32a).

The existing server object A 40 accessed by the client object 30 requests the management object 10 to acquire the server object information of the server object B accessed by the existing server object A 40.

Upon receipt of the request from the existing server object A 40 to acquire the server object information of the server object B, the management object 10 retrieves the server object information of the server object B, acquires the server object information of the existing server object B 50 already started as a server object B and supplies it to the requester existing server object A 40 (42).

The existing server object A 40 supplied with the server object information of the existing server object B 50 from the management object 10 accesses the existing server object B 50 using the particular server object information (43), and carries out the processing requested by the client object 30.

In the distributed system according to this embodiment, assume that the existing server object A 40 and the existing server object B 50 are to be changed. A new server object B 70 is started (24) and a new server object A 60 is started (25) by the factory object 20. The factory object 20 starts the new server object B 70 and the new server object A 60 in that order, i.e. in the order reverse to the predefined order of access of the server objects.

In the case where the client object 30 is executed in this distributed system of a new configuration, as in the previous case, the client object 30 requests the management object 10 to acquire the server object information of the sever object A accessed by the client object 30.

The management object 10, upon receipt of the request from the client object 30 to acquire the server object information of the server object A, retrieves the server object information of the server object A, compares the change information of the existing server object A 40 already started as a server object A with that of the new server object A 60, acquires the server object information of the new server object A 60 providing a new server object A and supplies it to the requester client object 30 (31).

The client object 30, upon receipt of the server object information of the new server object A 60 from the management object 10, accesses the new server object A 60 using the particular server object information (32b).

The new server object A 60 accessed by the client object 30 requests the management object 10 to acquire the server object information of the server object B accessed by the new server object A 60.

The management object 10, upon receipt of the request from the new server object A 60 to acquire the server object information of the server object B, retrieves the server object information of the server object B, compares the change information of the existing server object B 50 already started as a server object B with that of the new server object B 70, acquires the server object information of the new server object B 70 providing the new server object B, and supplies it to the requester ~~existing~~ new server object ~~A 40~~A 60 (62).

The new server object A 60 supplied with the server object information of the new server object B 70 from the management object 10 accesses the new server object B 70 using the particular server object information (63), and performs the process requested by the client object 30.

Also, in the distributed system of the new configuration, once the new server object A 60 and the new server object B 70 are started, the factory object 20 stops the existing server object B 50 (26) and the existing server object A 40 (27) as soon as the processes under execution by the existing server object ~~A 50~~B 50 and the existing server object A 40 are completed. The factory object 20 stops the existing server object B 50 and the existing server object A 40 in that order, i.e. in the predefined order reverse to the order of access of the server objects.

Fig. 2 is a diagram showing an example of configuration of the management object 10 according to an embodiment. As shown in Fig. 2, the management object 10 according to this embodiment includes a server object information management unit 10a and a server object information select unit 10b.

The server object information management unit 10a processes the registration requests from the server objects and manages the server object information of each server object. The server object information select unit 10b performs the process of selecting and supplying the requester with the server object information of the newest one of the server objects requested from the client object or the server object, and in the case where the requester is a server object, carries out the process for selecting and supplying the requester with the server object information of the server object having the change information concurrent with or older than the requester server object.

The program for causing the management object 10 to function as the server object information management unit 10a and the server object information select unit 10b is recorded in a recording medium such as a CD-ROM and stored in a magnetic disk or the like, after which it is loaded in a memory and executed. The recording medium for recording the program may be other than the CD-ROM.

Fig. 3 is a diagram showing an example of configuration of the factory object 20 according to an embodiment. As shown in Fig. 3, the factory object 20 according to this embodiment includes a server object life cycle management unit 20a. The server object life cycle management unit 20a is a processing unit for controlling the start and stop of the server objects, and in the presence of a plurality of server objects having the same ID information indicating the name of the server object and the interface, stopping the server objects having the old change information.

The program for causing the factory object 20 to function as the server object life cycle management unit 20a is recorded in a recording medium such as a CD-ROM and stored in a magnetic disk or the like, after which it is loaded in a memory and executed. The recording medium for recording this program may be other than the CD-ROM.

Fig. 4 is a diagram showing an example of configuration of the client object 30 according to an embodiment. As shown in Fig. 4, the client object 30 according to this embodiment includes a server object information acquisition unit 30a and a server object access unit 30b.

The server object information acquisition unit 30a is a processing unit for requesting the management object 10 to supply the server object information of the server object to be accessed. The server object access unit 30b is a processing unit for accessing the server object indicated in the server object information acquired from the management object 10.

The program for causing the client object 30 to function as the server object information acquisition unit 30a and the server object access unit 30b is recorded in a recording medium such as a CD-ROM and stored in a magnetic disk or the like, after which it is loaded in a memory and executed. The recording medium for recording the program may be other than the CD-ROM.

Fig. 5 is a diagram showing an example of configuration of a server object according to an embodiment. As shown in Fig. 5, the server object according to this embodiment includes a server object information registration unit 40a, a request processing unit 40b, another server object information acquisition unit 40c and another server object access unit 40d.

The server object information registration unit 40a is a processing unit for requesting the management object 10 to register a server object and sending the revision information indicating the newness of the particular server object as change information to the management object 10.

The request processing unit 40b is a processing unit for performing the process requested by the access from a client object or a server object. Another server object information acquisition unit 40c is a processing unit for inquiring of the management object 10 as to another server object which may be accessed by the server object. Another server object access unit 40d is a processing unit for accessing the another server object indicated in the server object information acquired from the management object 10.

The program for causing the server object to function as the server object information registration unit 40a, the request processing unit 40b, another server object information acquisition unit 40c and another server object access unit 40d is recorded in a recording medium such as the CD-ROM and stored in a magnetic disk or the like, after which it is loaded in a memory and executed. The recording medium for recording the program mentioned above may be other than the CD-ROM.

Fig. 6 is a diagram showing an example of configuration of the server object registration information used by the management object 10 in the server object information management unit 10a and the server object information select unit 10b according to this embodiment. As shown in Fig. 6, the server object registration information 100 of the management object 10 is configured with positional information 101 for storing the position of the server object, ID information 102 for storing the name of the server object and the interface information, and change information 103 for storing the information for recognizing the change of the server object.

The server object information management unit 10a of the management object 10, if supplied with the change information indicating the revision of a server object from the particular server object, registers the particular revision information in the change information 103, while in the case where the change information is not supplied to the server object information management unit 10a from the server object, registers in the change information 103 a flag indicating that the change information is not sent and the present time of registration of the particular server object.

Fig. 7 is a diagram showing an example of configuration of the server object management information used in the server object life cycle management unit 20a of the factory object 20 according to an embodiment. As shown in Fig. 7, the server object management information 200 of the factory object 20 is configured with positional information 201 for storing the position of the server object, ID information 202 for storing the name of the server object and the interface information, and change information 203 for storing information for recognizing the change in the server object.

Fig. 8 is a flowchart showing the steps of the processing in the server object information management unit 10a of the management object 10 according to this embodiment. Now, the registration process (41, 51, 61, 62) of the server objects (40, 50, 60, 70) by the server object information management unit 10a of the management object 10 according to the embodiment configured as described above will be explained with reference to the flowchart of Fig. 8.First, the information on the position of the server object is stored in the positional information 101, and the name of the server object and the information on the interface are stored in the ID information 102 (steps 301, 302).

Step 303 determines whether the change information 52 or 72 has been delivered or not from the server object, and in the case where the change information 52 or 72 has been so delivered, the change information 52 or 72, as the case may be, is stored in the change information 103 (step 304), while in the case where the change information 52 or 72 is not so delivered, on the other hand, a flag indicating that the change information is not delivered and the present time are stored (step 305).

Fig. 9 is a flowchart showing the steps of the processing in the server object information select unit 10b of the management object 10 according to an embodiment of the invention. Now, the process (31, 42, 62) for selecting the server objects (40, 50, 60, 70) by the server object information select unit 10b of the management object 10 according to the embodiment configured as described above will be explained with reference to the flowchart of Fig. 9.

First, step 401 retrieves the ID information 102 of the server object registration information 100 from the server object information delivered from the requester, and in the case where the selected server object is not registered, absence of a registered server object is notified to the requester (step 402).

Step 403 determines whether a plurality of the selected server objects are registered or not, and in the case where a plurality of them are not registered, the positional information 101 of the single server object in registration is selected and supplied to the requester (steps 404, 410).

Step 405 determines whether the requester is also registered as a server object or not, and in the case where it is not so registered, the positional information 101 of the server object having new change information 103 is selected and supplied to the requester (steps 406, 410). In the case where the revision information is registered in all the change information 103 of the server objects to be selected, the server object having the newest revision information is selected. In the case where the revision information is registered in some of the change information 103 of the server objects to be selected and a flag indicating that no change information is sent is registered in other change information 103, the server object having the newest revision information is selected from those with the change information in which the revision information is registered, while in the case where the flag indicating that the change information is not sent is registered in all the change information 103 of the server objects to be selected, the time information in the change information 103 selects the newest server object.

Step 407 determines whether there exists any server object having the same ID information 102 as the requester server object, and in the case where there is no other server object having the same ID information, the positional information 101 of the new server object having the same ID information 103 is selected and supplied to the requester (steps 406, 410).

By referring to the change information 103 of a server object having the same ID information 102 as the change information 103 of the requester server object, step 408 determines whether the particular server object is new or not, and in the case where it is new, the positional information 101 of a server object having the new change information 103 is selected and supplied to the requester (steps 406, 410). In the case where the server object is old, on the other hand, the positional information 101 of a server object having the old change information 103 is selected and supplied to the requester (steps 409, 410).

Fig. 10 is a flowchart showing the steps of the process performed in the server object life cycle management unit 20a of the factory object 20 according to an embodiment of this invention. Now, in this embodiment configured as described above, the start process (22, 23, 24, 25) and the stop process (26, 27) of the server object (40, 50, 60, 70) performed by the server object life cycle management unit 20a of the factory object 20 will be explained with reference to the flowchart of Fig. 10.

First, the server object life cycle management unit 20a of the factory object 20 starts the server objects in the order reverse to the predefined order of access to the server objects, so that the information on the position of the server objects is stored in the positional information 201 and the names of the server objects and the information on the interfaces are stored in the ID information 202 (steps 501 to 503).

Step 504 determines whether the change information 52 or 57 is delivered from a server object or not, and in the case where it is delivered so, the particular change information 52 or 57 is stored in the change information 203 (step 505), while in the case where the change information 52 or 57 is not so delivered, on the other hand, a flag indicating that the change information is not delivered and the present time are stored (step 506).

In the case where step 507 determines that a plurality of server objects (40, 60; 50, 70) have the same ID information 202, the change information 203 thereof are compared with each other. Any object (40, 50) of which the change information 203 is found to be old is stopped (step 508). In the process, as soon as the process under execution and having the old object (40, 50) is completed, the server object life cycle management unit 20a of the factory object 20 stops the old objects (40, 50) in the order reverse to the predefined order of access.

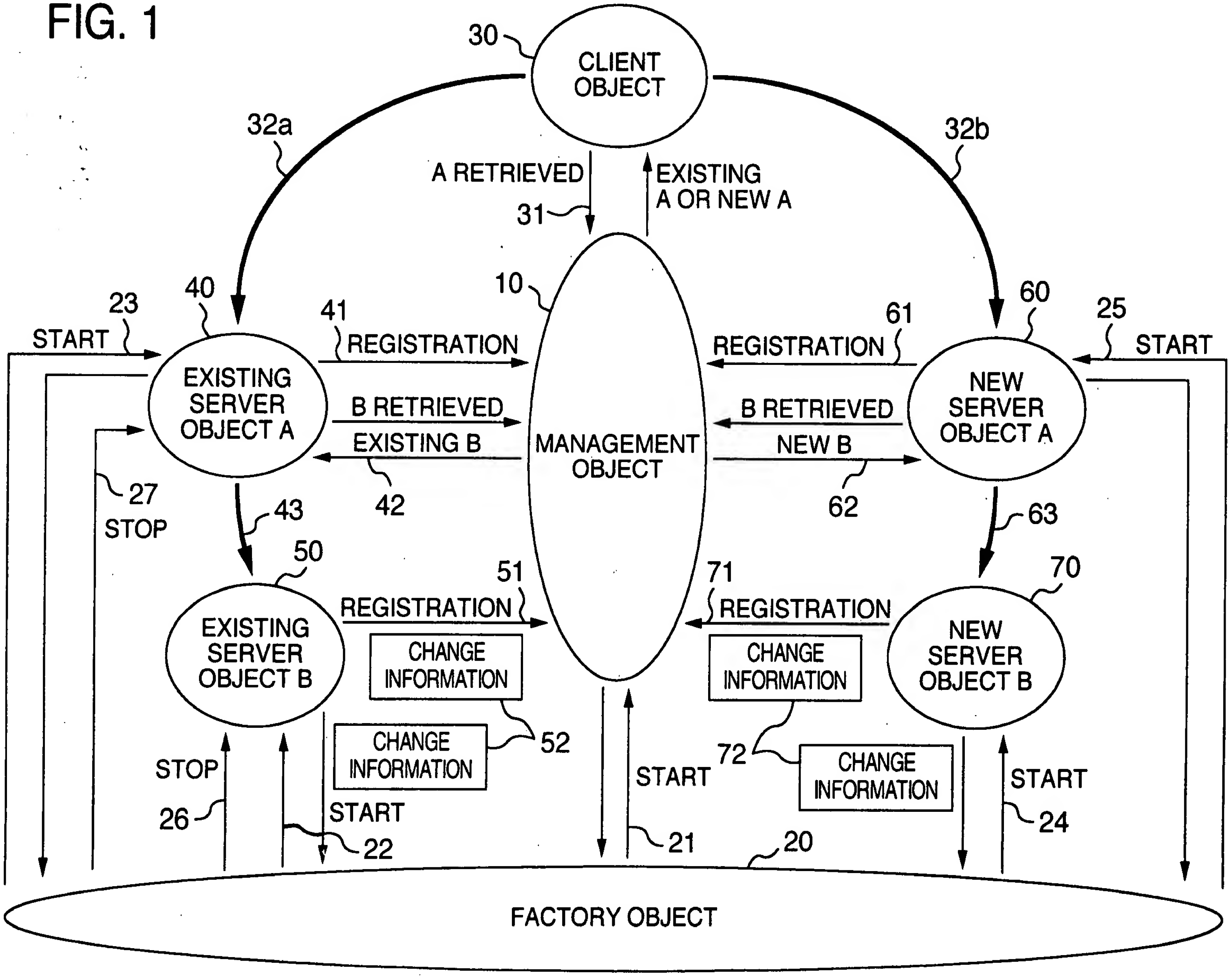
As described above, in the distributed system according to this embodiment, the information that can be determined by the management object and the factory object is held as change information, and according to the difference between these change information, the change of a server object is identified, thereby making it possible to change an object in operation.

It will be understood from the foregoing description that with the distributed object management system according to this embodiment, the server object information of the newest server object is supplied to the requester and the newest server object is accessed in the case where the server object information is requested. In this way, the server object accessed is changed from the old to the newest one, and therefore the server object can be changed in operation.

According to this invention, in the case where the server object information is requested, the server object information of the newest server object is supplied to the requester thereby to access the newest server object. In this way, the server object accessed can be changed from the old to the newest one, and therefore the server object can be changed in operation.

ANNOTATED MARKED-UP DRAWING

FIG. 1



ANNOTATED MARKED-UP DRAWING

FIG. 2 Prior Art

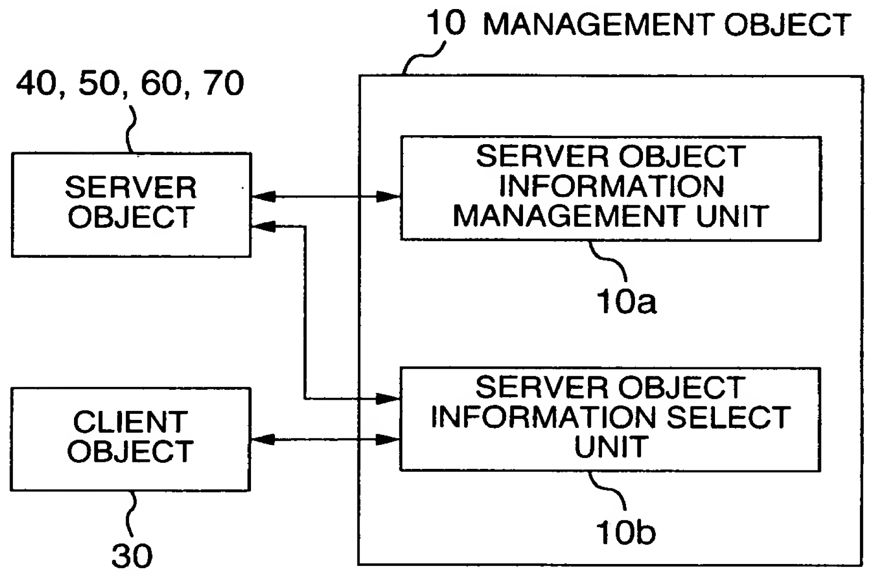
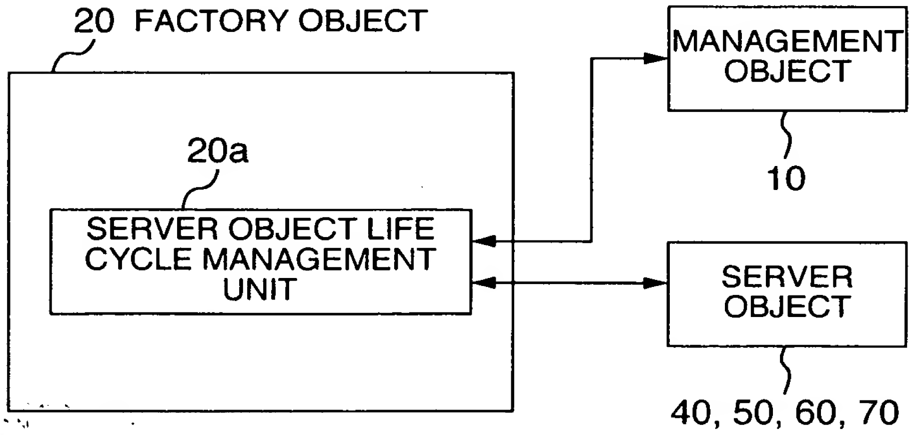
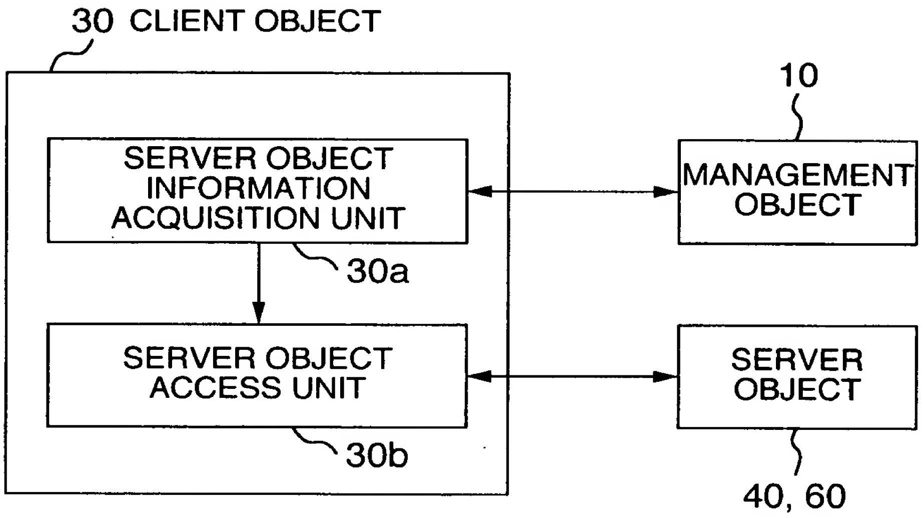

10 MANAGEMENT OBJECT
40, 50, 60, 70
SERVER OBJECT
SERVER OBJECT INFORMATION MANAGEMENT UNIT
10a
CLIENT OBJECT
SERVER OBJECT INFORMATION SELECT UNIT
30
10b

FIG. 3 Prior Art



ANNOTATED MARKED-UP DRAWING

FIG. 4 Prior Art


30 CLIENT OBJECT
SERVER OBJECT INFORMATION ACQUISITION UNIT
30a
SERVER OBJECT ACCESS UNIT
30b
10
MANAGEMENT OBJECT
SERVER OBJECT
40, 60